🔐 Tutoriel : Intégration de Fail2Ban avec Wazuh

By Ludovic Gildas DOAMBA

LinkedIn https://www.linkedin.com/in/ludovic-g-doamba-45960591/

🎯 Objectif

Mettre en place une intégration native entre Fail2Ban et Wazuh pour :

- 🚨 Détecter les attaques bruteforce SSH
- 👁️ Afficher les alertes dans Wazuh Dashboard
- 🛡️ Exploiter un SOC souverain sans licence propriétaire
- Permettre la **corrélation avec d'autres événements de sécurité**

🧭 Introduction

Fail2Ban est un outil open source de protection contre les attaques par force brute, en particulier sur les services comme SSH. Il analyse les logs pour détecter les échecs de connexion répétés et bannit automatiquement les IP concernées.

Wazuh, de son côté, est un SIEM open source complet qui permet de collecter, analyser, corréler et visualiser les événements de sécurité.

L'intégration des deux permet de :

- visualiser les IP bannies directement dans Kibana,
- enrichir la détection des menaces,
- maintenir une solution de cybersécurité 100% souveraine sans dépendance à des éditeurs propriétaires.

En intégrant Fail2Ban à Wazuh, vous :

✅ Centralisez les bannissements Fail2Ban dans votre SIEM
✅ Bénéficiez d'une vue corrélée sur les menaces
✅ Automatisez les réponses et alertes
✅ Renforcez la **souveraineté** de votre infrastructure

Prérequis Matériels & Logiciels

Matériel minimal recommandé pour fail2ban (pour un petit environnement de test) :

- **CPU** : 2 vCPU
- **RAM** : 2 Go minimum
- **Stockage** : 10 Go libres
- **Accès root** ou compte avec sudo
- **Connexion Internet** pour installer les paquets

Logiciels :

- **Debian 12/Ubuntu 22.04 ou +** pour Fail2Ban
- **Wazuh Agent** installé sur la machine **Debian** ou **Ubuntu**
- Un **serveur Wazuh (Wazuh manager)** opérationnel, veuillez suivre ce Guide d'installation Wazuh, pour déployer le Serveur **Wazuh** et ses différents agents.

🧪 Étape 1 — Installer Fail2Ban

Installation sur Debian/Ubuntu : →

sudo apt update && sudo apt install fail2ban -y

```
root@PC-Deb12:~# apt update
Atteint :1 http://deb.debian.org/debian bullseye InRelease
Atteint :2 http://security.debian.org/debian-security bullseye-security InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
```

```
root@PC-Deb12:~# apt install fail2ban -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  python3-pyinotify python3-systemd whois
Paquets suggérés :
  mailx monit sqlite3 python-pyinotify-doc
Les NOUVEAUX paquets suivants seront installés :
  fail2ban python3-pyinotify python3-systemd whois
0 mis à jour, 4 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 596 ko dans les archives.
Après cette opération, 2□819 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bullseye/main amd64 fail2ban all 0.11.2-2 [451 kB]
Réception de :2 http://deb.debian.org/debian bullseye/main amd64 python3-pyinotify all 0.9.6-1.3 [27,2 kB]
Réception de :3 http://deb.debian.org/debian bullseye/main amd64 python3-systemd amd64 234-3+b4 [36,4 kB]
Réception de :4 http://deb.debian.org/debian bullseye/main amd64 whois amd64 5.5.10 [81,1 kB]
596 ko réceptionnés en 0s (5□589 ko/s)
Sélection du paquet fail2ban précédemment désélectionné.
(Lecture de la base de données... 34264 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../fail2ban_0.11.2-2_all.deb ...
Dépaquetage de fail2ban (0.11.2-2) ...
Sélection du paquet python3-pyinotify précédemment désélectionné.
Préparation du dépaquetage de .../python3-pyinotify_0.9.6-1.3_all.deb ...
Dépaquetage de python3-pyinotify (0.9.6-1.3) ...
Sélection du paquet python3-systemd précédemment désélectionné.
Préparation du dépaquetage de .../python3-systemd_234-3+b4_amd64.deb ...
Dépaquetage de python3-systemd (234-3+b4) ...
Sélection du paquet whois précédemment désélectionné.
Préparation du dépaquetage de .../whois_5.5.10_amd64.deb ...
Dépaquetage de whois (5.5.10) ...
Paramétrage de whois (5.5.10) ...
Paramétrage de fail2ban (0.11.2-2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /lib/systemd/system/fail2ban.service.
Paramétrage de python3-pyinotify (0.9.6-1.3) ...
Paramétrage de python3-systemd (234-3+b4) ...
Traitement des actions différées (« triggers ») pour man-db (2.9.4-2) ...
root@PC-Deb12:~#
```

Rôle des fichiers par défaut

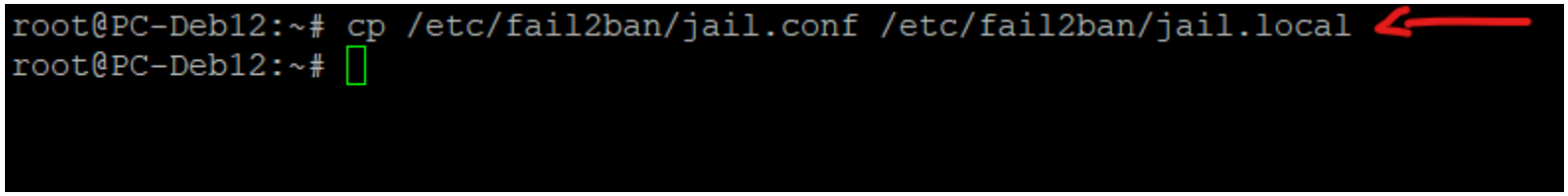
Lors de l'installation, deux fichiers de configuration principaux sont créés :

- **/etc/fail2ban/jail.conf**
  Contient la configuration générale de Fail2Ban (toutes les jails, paramètres par défaut).
- **/etc/fail2ban/jail.d/defaults-debian.conf**
  Spécifique à Debian/Ubuntu, contient les jails activées par défaut selon la distribution.

⚠️ **Important** : Ces fichiers **ne doivent pas être modifiés directement**, car toute mise à jour de Fail2Ban pourrait les écraser.

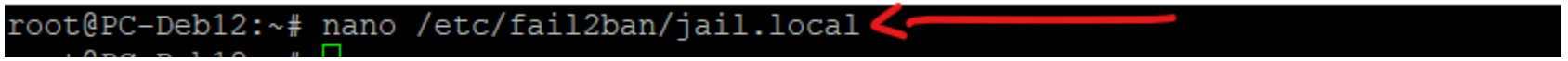
Création du fichier de configuration personnalisé jail.local

sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local



Toutes les personnalisations se font dans **/etc/fail2ban/jail.local**.

Editer le fichier Configuration de base jail.local



Modification des paramètres clés (mon cas)

Dans mon environnement, j'ai modifié :

Paramètre	Valeur configurée	Description
bantime	5m	Durée du bannissement (5 minutes)
findtime	5m	Fenêtre de temps pour comptabiliser les échecs
maxretry	5	Nombre de tentatives avant bannissement
ignoreip	IP(s) en liste blanche	Adresses IP à ne pas bannir

```
# "ignoreself" specifies whether the local resp. own IP addresses should be ignored
# (default is true). Fail2ban will not ban a host which matches such addresses.
#ignoreself = true

# "ignoreip" can be a list of IP addresses, CIDR masks or DNS hosts. Fail2ban
# will not ban a host which matches an address in this list. Several addresses
# can be defined using space (and/or comma) separator.
#ignoreip = 127.0.0.1/8 ::1

# External command that will take an tagged arguments to ignore, e.g. <ip>,
# and return true if the IP is to be ignored. False otherwise.
#
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned.
bantime  = 5m

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime  = 5m

# "maxretry" is the number of failures before a host get banned.
maxretry = 5

# "maxmatches" is the number of matches stored in ticket (resolvable via tag <matches> in
maxmatches = %(maxretry)s

# "backend" specifies the backend used to get files modification.
# Available options are "pyinotify", "gamin", "polling", "systemd" and "auto".
# This option can be overridden in each jail as well.
#
# pyinotify: requires pyinotify (a file alteration monitor) to be installed.
#              If pyinotify is not installed, Fail2ban will use auto.
# gamin:     requires Gamin (a file alteration monitor) to be installed.
#              If Gamin is not installed, Fail2ban will use auto.
# polling:   uses a polling algorithm which does not require external libraries.
# systemd:   uses systemd python library to access the systemd journal.
#              Specifying "logpath" is not valid for this backend.
#              See "journalmatch" in the jails associated filter config
# auto:      will try to use the following backends, in order:
#              pyinotify, gamin, polling.
#
```

🔐 Paramètres clés à connaître

Dans Fail2ban il y'a des directives importantes à connaitre, dans cette capture ci-dessous je vais vous parler de : bantime, findtime, maxretry

- **bantime :** La durée de ban d'une adresse IP (Je l'ai fixé à 5m ici)
- **findtime :** La durée pour comptabiliser le nombre de tentatives maximales avant de déclencher un bannissement
- **maxretry :** Le nombre de tentative durant la période indiquée dans findtime avant de faire un bannissement .

NB : Dans ma configuration actuelle, si 5 échecs d'authentification sont détectés dans un intervalle de 5 minutes, l'adresse IP sera banni pendant 5minutes.

Ignoreip : Décommenter cette ligne si vous souhaitez ajouter les adresses IPs à ne pas bannir (Une liste blanche)

```
  GNU nano 5.4                                                    /etc/fail2ban/jail.local *
# "ignoreself" specifies whether the local resp. own IP addresses should be ignored
# (default is true). Fail2ban will not ban a host which matches such addresses.
#ignoreself = true

# "ignoreip" can be a list of IP addresses, CIDR masks or DNS hosts. Fail2ban
# will not ban a host which matches an address in this list. Several addresses
# can be defined using space (and/or comma) separator.
#ignoreip = 127.0.0.1/8 ::1
```

Activer la surveillance SSH dans /etc/fail2ban/jail.local :

```
#
# [sshd]
enabled = true
#
# See jail.conf(5) man page for more information
```

[sshd]

enabled = true

port = ssh

maxretry = 5

Activer et démarrer le service :

sudo systemctl enable fail2ban

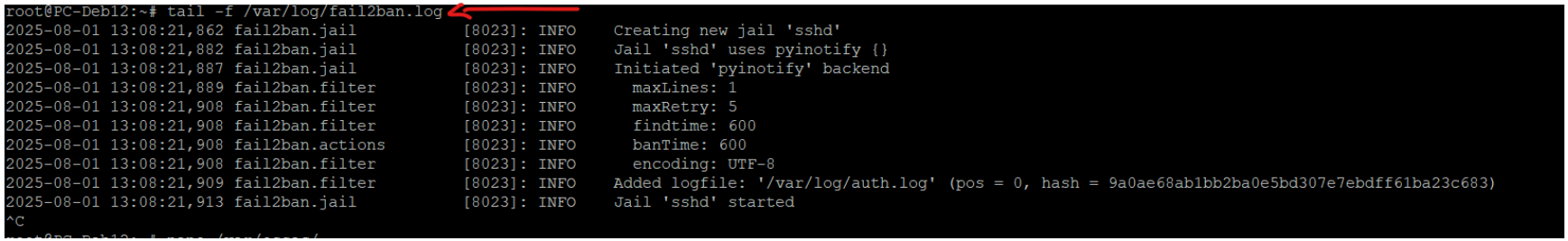
sudo systemctl start fail2ban

```
root@PC-Deb12:~# systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
root@PC-Deb12:~# systemctl start fail2ban
root@PC-Deb12:~# systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2025-08-01 13:08:21 CEST; 14min ago
       Docs: man:fail2ban(1)
   Main PID: 8023 (fail2ban-server)
      Tasks: 5 (limit: 4618)
     Memory: 17.1M
        CPU: 562ms
     CGroup: /system.slice/fail2ban.service
             └─8023 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

août 01 13:08:21 PC-Deb12 systemd[1]: Starting Fail2Ban Service...
août 01 13:08:21 PC-Deb12 systemd[1]: Started Fail2Ban Service.
août 01 13:08:21 PC-Deb12 fail2ban-server[8023]: Server ready
root@PC-Deb12:~#
```

Localisation des logs Fail2Ban

Tous les événements Fail2Ban sont enregistrés dans :

/var/log/fail2ban.log



Exemple de log :

2025-08-01 14:26:22,755 fail2ban.actions [8023]: NOTICE [sshd] Ban 192.168.1.15

Étape 2— Configurer l'agent Wazuh pour lire les logs de Fail2ban

Dans le fichier **/var/ossec/etc/ossec.conf**, ajouter :

```
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/fail2ban.log</location>
</localfile>
```

```
GNU nano 5.4                                                    /var/ossec/etc/ossec.conf
    </synchronization>
  </syscheck>

  <!-- Log analysis -->
  <localfile>
    <log_format>command</log_format>
    <command>df -P</command>
    <frequency>360</frequency>
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/fail2ban.log</location>
  </localfile>
```

Explication :

- <log_format>syslog</log_format> : Indique que le fichier suit un format type syslog.
- <location> : Chemin exact du fichier à surveiller.

Puis redémarrer l'agent Wazuh :

sudo systemctl restart wazuh-agent

```
root@PC-Deb12:~# systemctl restart wazuh-agent
```

🧩 Étape 3— Créer un décodeur personnalisé Wazuh

📌 Par défaut, **fail2ban_decoder.xml** n'existe pas. Il faut le **créer** dans : /var/ossec/etc/decoders/**fail2ban_decoder.xml**

```
<decoder name="fail2ban">

     <prematch>[\d+]:\s*\w+\s*[\w*]</prematch>

</decoder>

<decoder name="fail2ban">

    <parent>fail2ban</parent>

   <regex>[(\d+)]:\s*(\w+)\s*[(\w*)]\s*(\w*)\s*(\d+.\d+.\d+.\d+)$</regex>

  <order>process_id,log_level,jail,action,srcip</order>

</decoder>
```

fail2ban_decoder.xml

```
<!-- Modify it at your will. -->
<decoder name="fail2ban">
    <prematch>[\d+]:\s*\w+\s*[\w*]</prematch>
</decoder>

<decoder name="fail2ban">
    <parent>fail2ban</parent>
    <regex>[(\d+)]:\s*(\w+)\s*[(\w*)]\s*(\w*)\s*(\d+.\d+.\d+.\d+)$</regex>
    <order>process_id,log_level,jail,action,srcip</order>
</decoder>

```

📜 Étape 4— Créer la règle d'alerte dans Wazuh

📌 Par défaut**, fail2ban_rule.xml** n'existe pas non plus. Il faut le créer dans **: /var/ossec/etc/rules/fail2ban_rule.xml**

```
<group name="fail2ban,">
  <rule id="100200" level="5">
    <decoded_as>fail2ban</decoded_as>
    <action>Ban</action>
    <description>Fail2Ban has banned an IP address: $(srcip) from jail: $(jail)</description>
    <group>authentication_failure,</group>
    </rule>
</group>
<group name="fail2ban,">
  <rule id="100201" level="5">
    <decoded_as>fail2ban</decoded_as>
    <action>Unban</action>
    <description>Fail2Ban: IP $(srcip) unbanned from $(jail)</description>
    <group>authentication_success,</group>
```

```
    </rule>

</group>
```

fail2ban_rule.xml

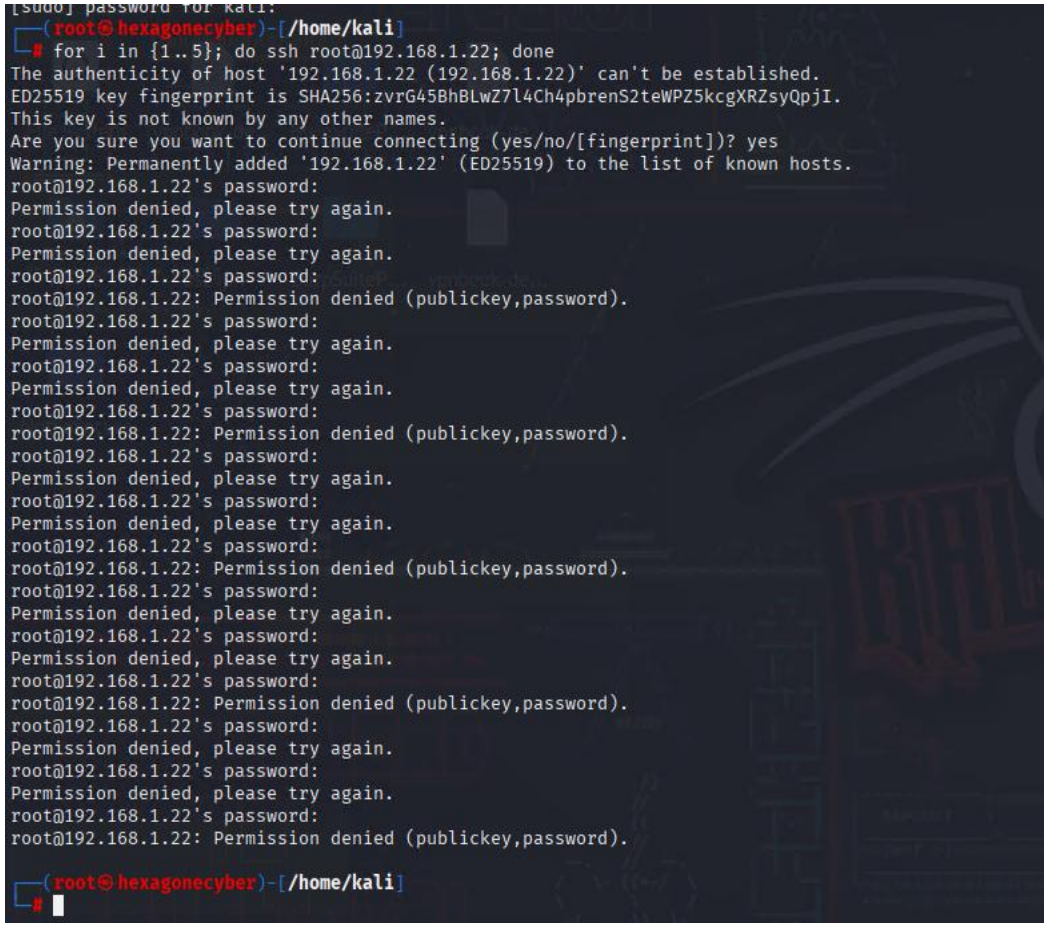
```
<!-- Modify it at your will. -->
<group name="fail2ban,">
  <rule id="100200" level="5">
    <decoded_as>fail2ban</decoded_as>
    <action>Ban</action>
    <description>Fail2Ban has banned an IP address: $(srcip) from jail: $(jail)</description>
    <group>authentication_failure,</group>
  </rule>
</group>

<group name="fail2ban,">
  <rule id="100201" level="5">
    <decoded_as>fail2ban</decoded_as>
    <action>Unban</action>
    <description>Fail2Ban: IP $(srcip) unbanned from $(jail)</description>
    <group>authentication_success,</group>
  </rule>
</group>

```

NB : Assurez-vous de respecter la syntaxe XML exacte (aucune balise non fermée, pas d'espaces indésirables).

Redémarrez le service Wazuh pour appliquer les changements (systemctl restart wazuh-manager)

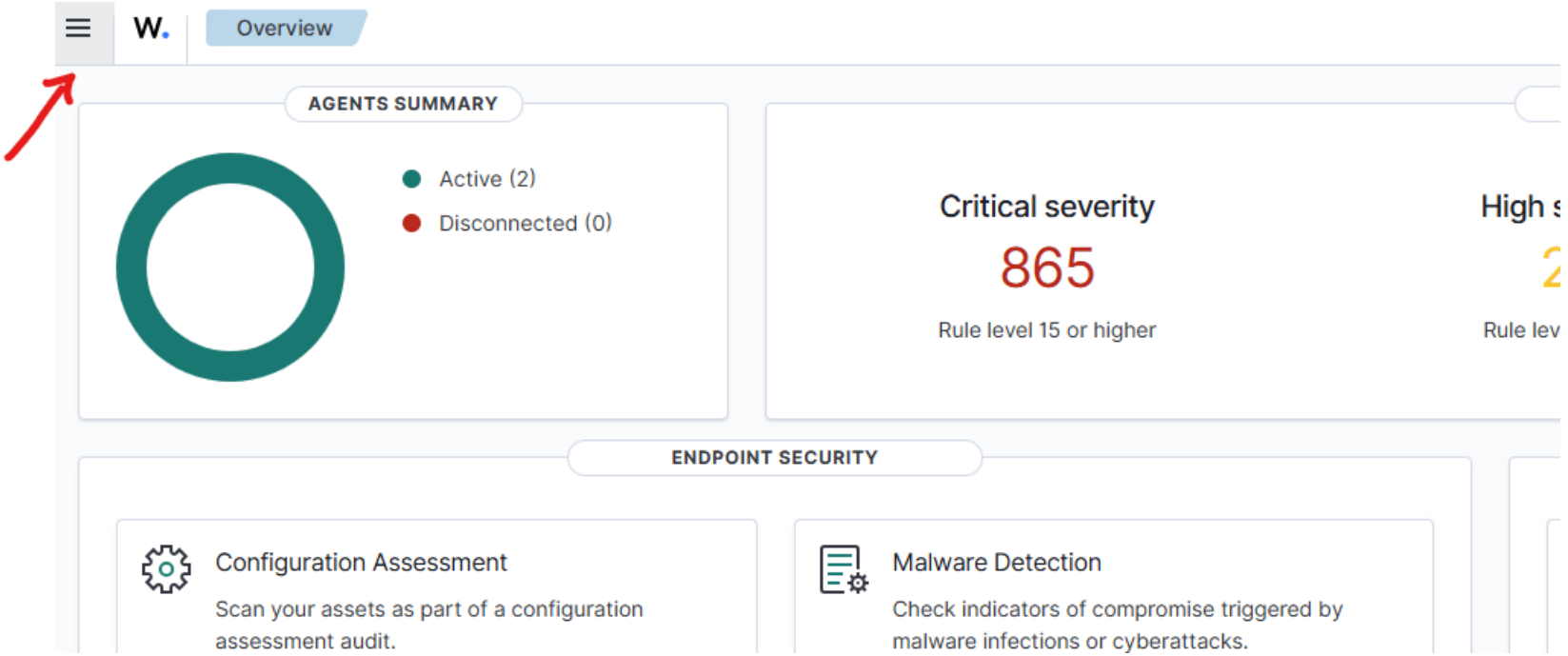
💣 Étape 5— Simuler une attaque SSH

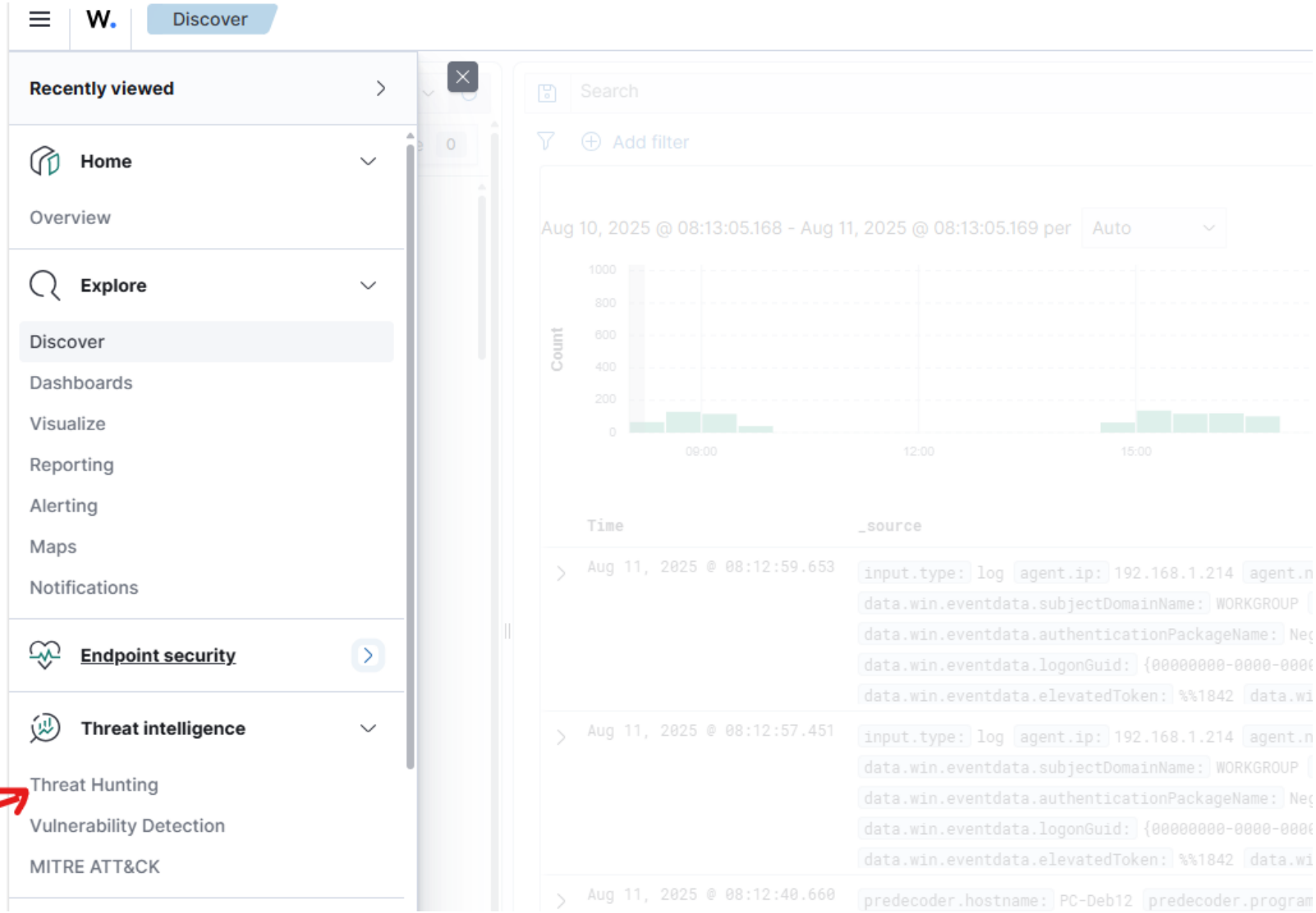
Depuis une machine Kali ou autre machine distante : for i in {1..5}; do ssh root@192.168.1.22; done



📊 Étape 6— Visualiser les alertes dans Wazuh Dasboard

Les alertes apparaissent avec la **rule.id** correspondante (100200 = Ban, 100201 = Unban)

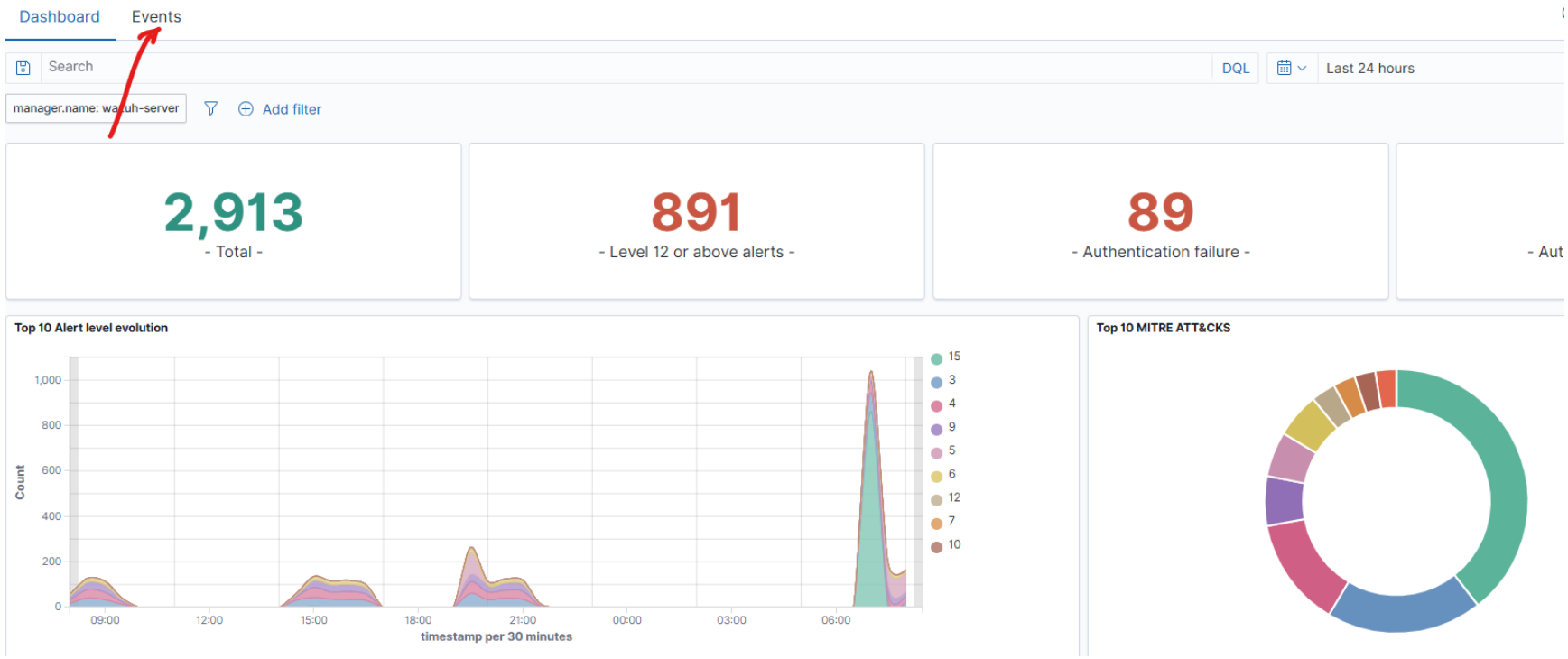



Discover
Recently viewed
Home
Overview
Explore
Discover
Dashboards
Visualize
Reporting
Alerting
Maps
Notifications
Endpoint security
Threat intelligence
Threat Hunting
Vulnerability Detection
MITRE ATT&CK
Search
Add filter
Aug 10, 2025 @ 08:13:05.168 - Aug 11, 2025 @ 08:13:05.169 per
Auto
Count
1000
800
600
400
200
0
09:00
12:00
15:00
Time
_source
Aug 11, 2025 @ 08:12:59.653
input.type: log agent.ip: 192.168.1.214
data.win.eventdata.subjectDomainName: WORKGROUP
data.win.eventdata.logonGuid: {00000000-0000-0000
data.win.eventdata.elevatedToken: %%1842
Aug 11, 2025 @ 08:12:57.451
input.type: log agent.ip: 192.168.1.214
data.win.eventdata.subjectDomainName: WORKGROUP
data.win.eventdata.logonGuid: {00000000-0000-0000
data.win.eventdata.elevatedToken: %%1842
Aug 11, 2025 @ 08:12:40.660
predecoder.hostname: PC-Deb12

HEXAGONE CYBER

By Ludovic Gildas DOAMBA

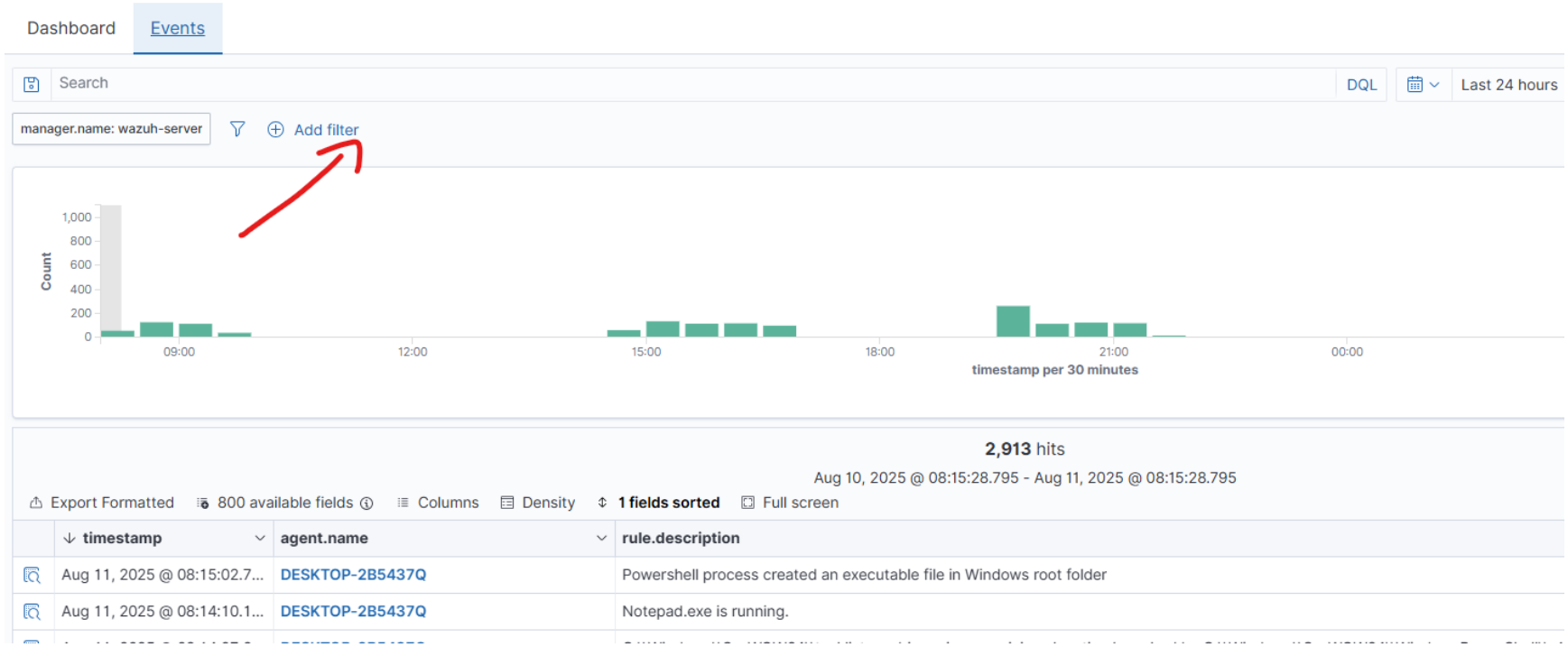
wazuh.

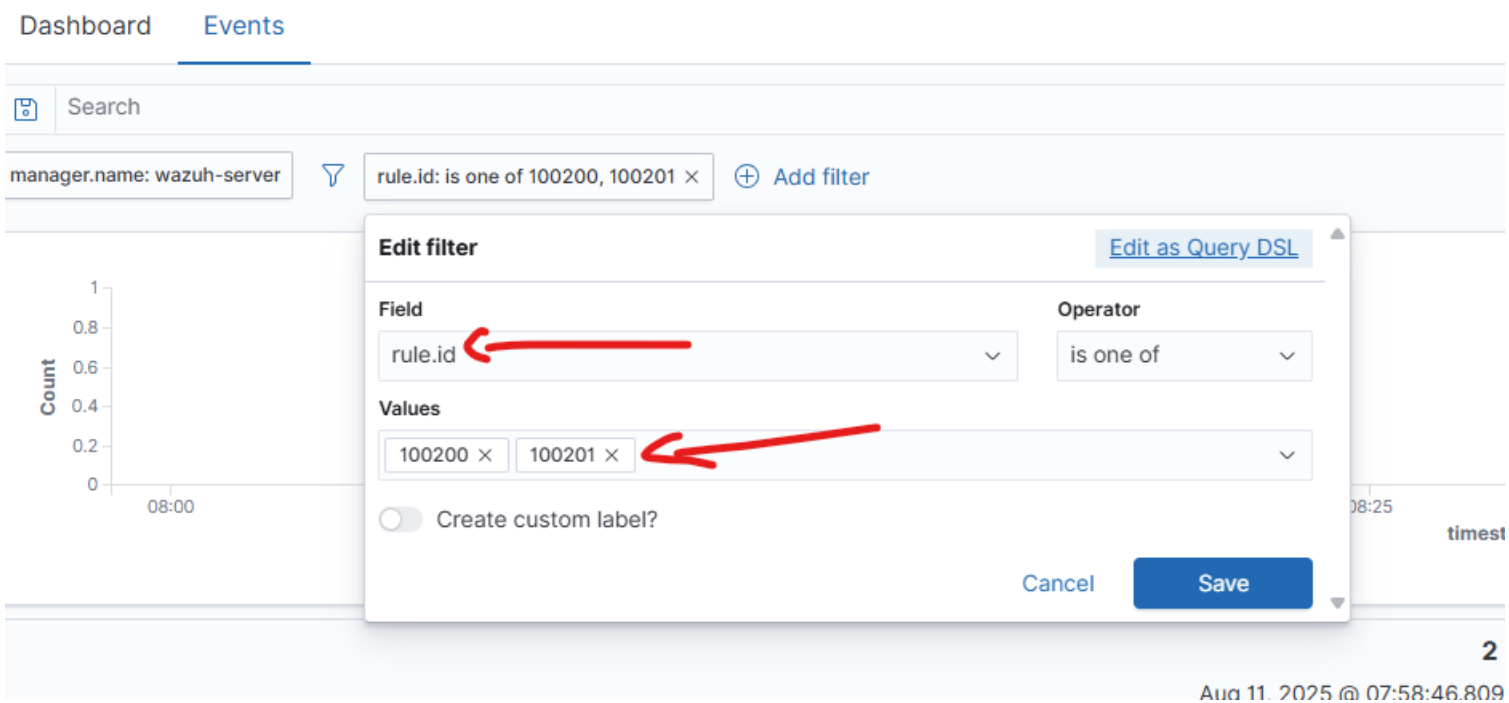


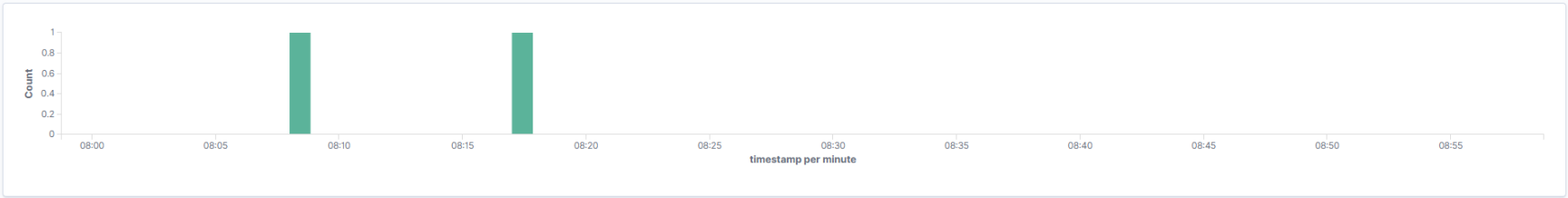
HEXAGONE CYBER

By Ludovic Gildas DOAMBA

wazuh.




Dashboard
Events
Search
manager.name: wazuh-server
rule.id: is one of 100200, 100201
Add filter
Edit filter
Edit as Query DSL
Field
rule.id
Operator
is one of
Values
100200
100201
Create custom label?
Cancel
Save
Count
1
0.8
0.6
0.4
0.2
0
08:00
08:25
timest
2
Aug 11, 2025 @ 07:58:46.809



2 hits

Aug 11, 2025 @ 07:58:46.809 - Aug 11, 2025 @ 08:58:46.810

Export Formatted | 800 available fields | Columns | Density | **1 fields sorted** | Full screen

	timestamp	agent.name	rule.description	rule.level	rule.id
	Aug 11, 2025 @ 08:17:38.2...	PC-Deb12	Fail2Ban: IP 192.168.1.15 unbanned from sshd	5	100201
	Aug 11, 2025 @ 08:08:23.2...	PC-Deb12	Fail2Ban has banned an IP address: 192.168.1.15 from jail: sshd	5	100200

t	agent.id	002
t	agent.ip	192.168.1.22
t	agent.name	PC-Deb12
t	data.action	Ban
t	data.jail	sshd
t	data.log_level	NOTICE
t	data.process_id	8023
t	data.srcip	192.168.1.15
t	decoder.name	fail2ban
t	full_log	2025-08-01 14:26:22,755 fail2ban.actions [8023]: NOTICE [sshd] Ban 192.168.1.15
t	id	1754051184.6782212
t	input.type	log
t	location	/var/log/fail2ban.log
t	manager.name	wazuh-server
t	predecoder.timestamp	2025-08-01 14:26:22,755
t	rule.description	Fail2Ban has banned an IP address: 192.168.1.15 from jail: sshd
#	rule.firedtimes	1
t	rule.groups	fail2ban, authentication_failure
t	rule.id	100200
#	rule.level	5

Conclusion :

Grâce à cette intégration :

- Vous centralisez les bannissements **Fail2Ban** dans un **SIEM souverain**
- Vous renforcez vos capacités de **détection des menaces.**
- Vous pouvez déclencher des réponses automatiques depuis **Wazuh**
- Vous supprimez les silos entre logs systèmes, applicatifs et sécurité

Ce tutoriel vous offre une base pour industrialiser vos mécanismes de protection tout en maintenant un niveau de souveraineté élevé sur votre cybersécurité.

📌 Résumé des bénéfices

- 🔍 **Détection brute-force** : via Fail2Ban
- 📌 **Décodage XML personnalisé** : fail2ban_decoder.xml
- 🧠 **Règles de corrélation** : fail2ban_rule.xml
- 📈 **Visualisation** : Wazuh Dashboard

🔗 À propos de l'auteur

Je suis **Ambassadeur Wazuh**, spécialisé en cybersécurité souveraine et open source.

👉 **Besoin d'un SOC souverain ?** Contactez-moi pour un accompagnement personnalisé.